# Informative Neural Ensemble Kalman Learning

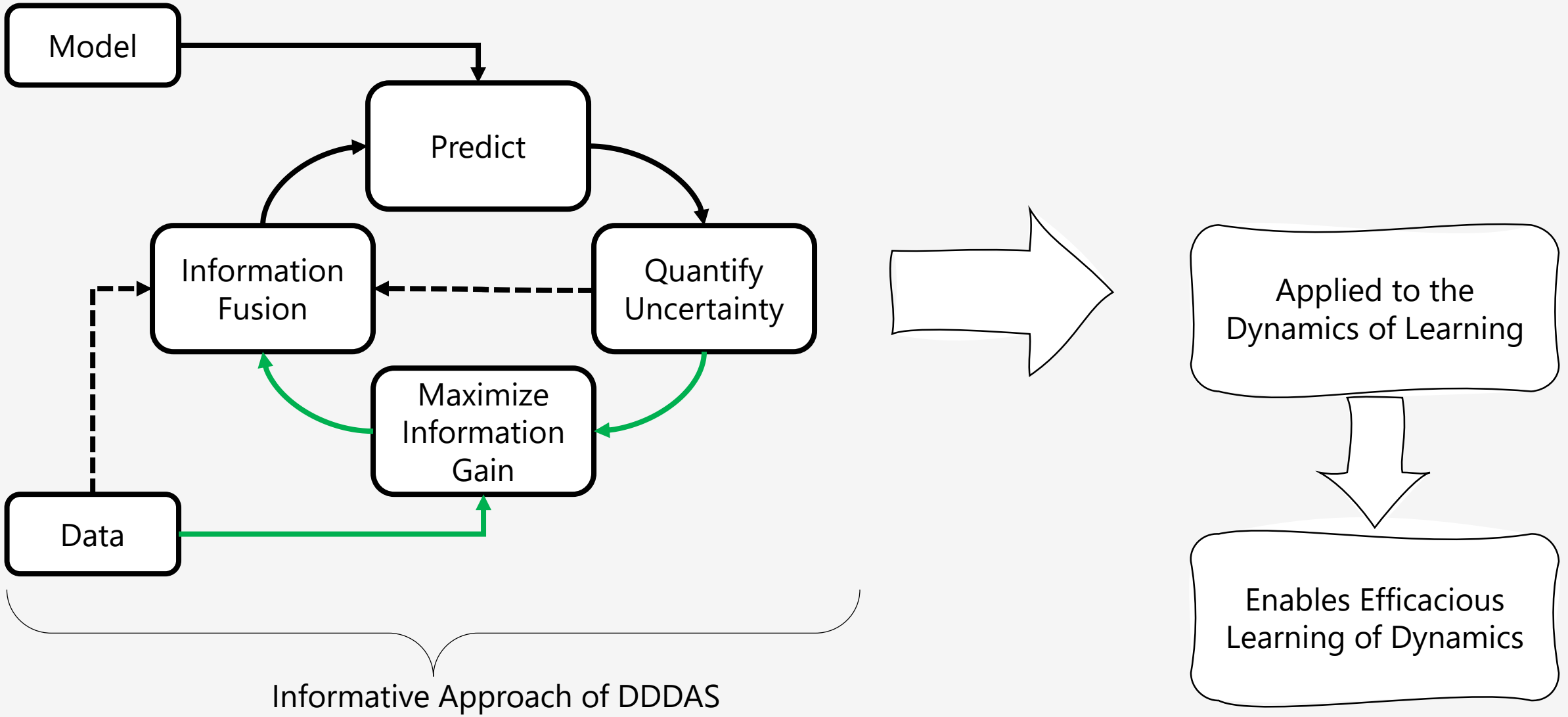
Margaret Trautner Gabriel Margolis Sai Ravela

Earth Signals and Systems Group
Earth, Atmospheric and Planetary Sciences
Massachusetts Institute of Technology, Cambridge, MA
http://essg.mit.edu

# Outline

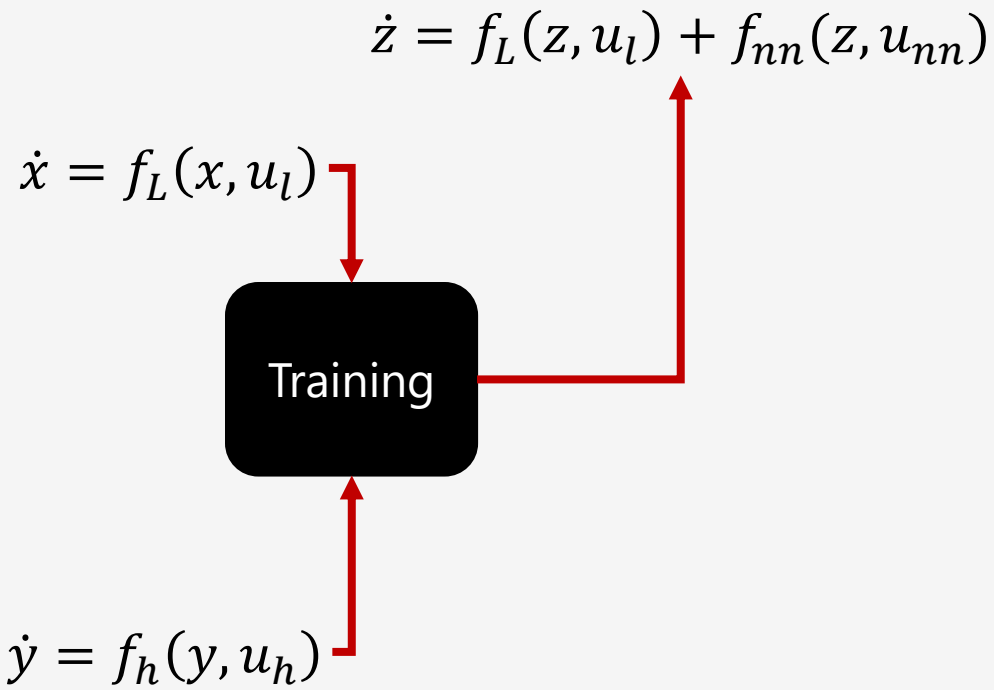
- The DDDAS Paradigm
- DDDAS for Informative Learning
- Informative Learning to discover dynamics from data
  - Jointly optimizing neural structure and parameters

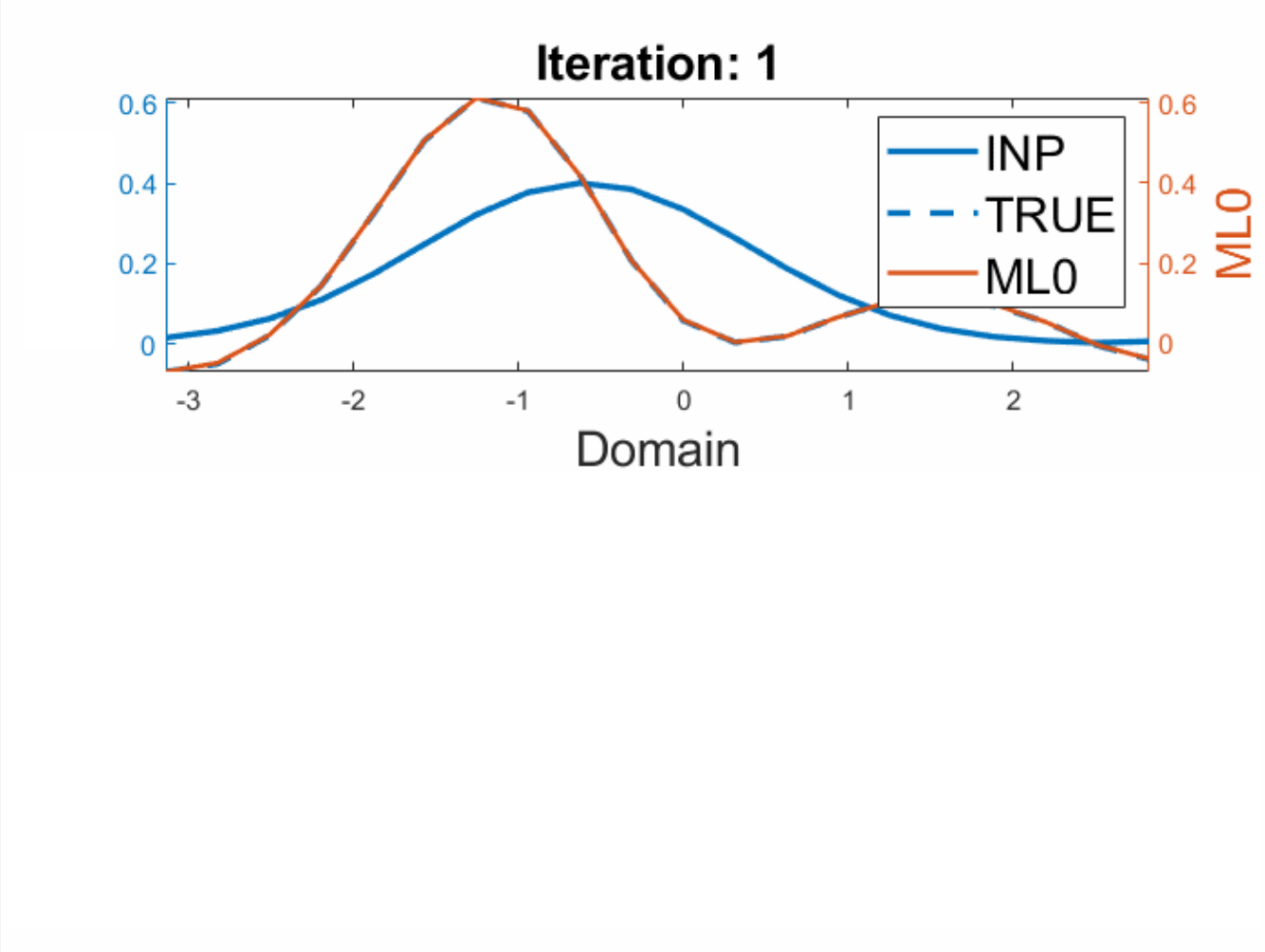
# Dynamic Data Driven Learning Systems



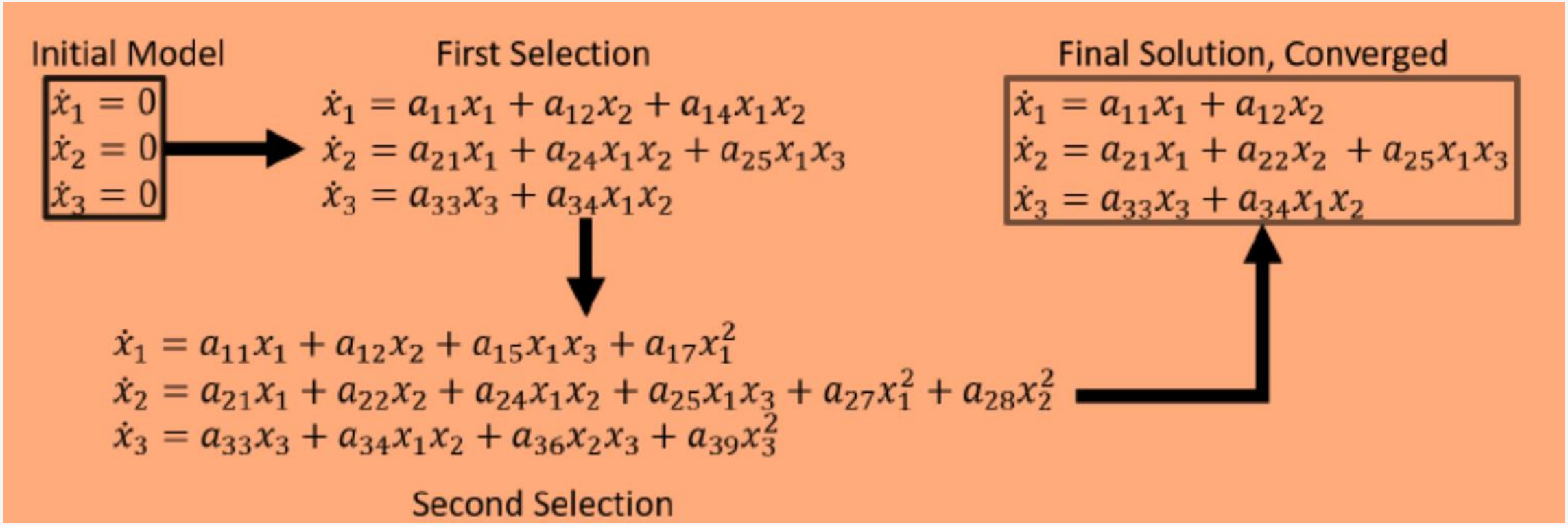

Also see: **arXiv:2008.09915**

# Stability, Tractability – motivation for dynamic data driven approach



# Learning Dynamics from Data



# Neural Networks as Dynamical Systems

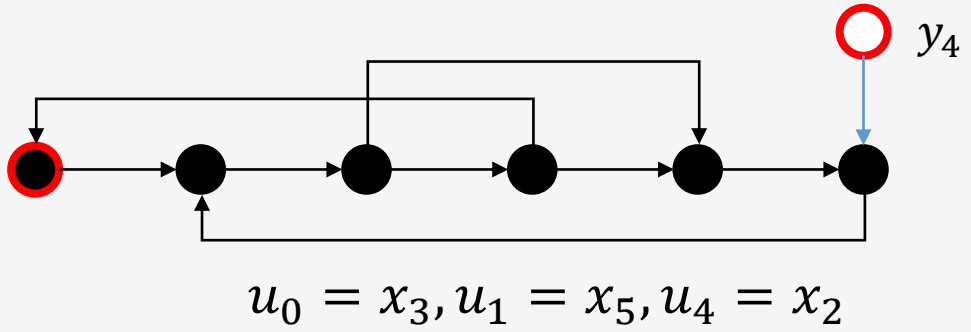
The Network

$$x_N = F_{NN}(x_0; \alpha) \quad (1)$$

Multistage Process

$$x_{l+1} = F_l(x_l, u_l; \alpha_l), 0 \le l < N \quad (2)$$

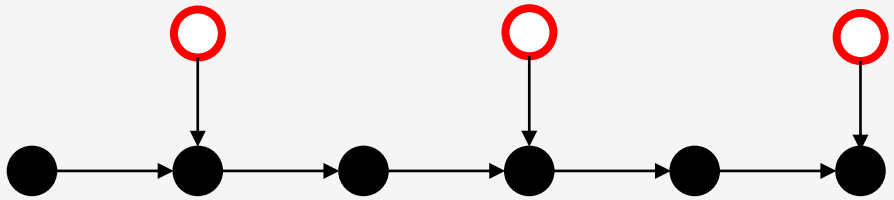
$$y_N = x_N + \nu_N,\ \nu_N \sim \mathcal{N}(0, R) \quad (3)$$




Neural Dynamical System

$$x_{n+1} = F(x_n, u_n; \alpha), \quad (4)$$

$$y_n = h(x_n) + \nu_n,\ \nu_n \sim \mathcal{N}(0, R_n) \quad (5)$$



# The Dynamics of Learning

## Objective

$$J(\cdot;[x_0,y_N]_s) \coloneqq \frac{1}{2}(y_N - x_N)^T R_N^{-1}(y_N - x_N) + \sum_{l=1}^{N} \gamma_l^T \{x_l - F_{l-1}(x_{l-1};\alpha_{l-1}) \quad (6)$$

## Forward-Backward

$$Input: \quad x_0 := [x_0]_s \quad (7)$$

$$Forward: \quad x_l = F_{l-1}(x_{l-1};\alpha_{l-1}), 0 < l \leq N \quad (8)$$

$$Terminal\ Error: \quad \gamma_N = R_N^{-1}([y_N]_s - x_N) \quad (9)$$

$$Backward: \quad \gamma_k = \left(\nabla_{x_k} F_k\right)^T \gamma_{k+1} \quad (10)$$

$$Parameter\ Gradient: \quad \frac{\partial J}{\partial \alpha_k} = \left(\nabla_{\alpha_k} F_k\right)^T \gamma_{k+1} \quad (11)$$

## Gradient Descent

$$\Delta\alpha_l = \frac{1}{S}\sum_{s=1}^{S} \frac{\partial}{\partial \alpha_k} J(\alpha_l;[x_0,y_N]_s) \quad (12)$$

To further examine the stochastic dynamics of learning, please see: [arXiv:2008.09915]

# An Ensemble Approach to Deep Learning

Parameter Estimation

$A_i$ : *parameter ensemble at iteration* $i > 0$
$X_{i,s}$: *ensemble of Network Predictions using training input* $x_{1,s}$
$Y_s$ : *ensemble of targets from traning output* $y_N, s$
$B_i$ : *minibatch at iteration* $i$ *of size* $S_i$
$\tilde{X}$ : *ensemble of deviations from mean vector*

$$\mathrm{A_{i+1}} = \mathrm{A_i} \frac{1}{S_i} \sum_{s \in B_i} \tilde{X}_{i,s}^T \left( \tilde{X}_{i,s} \tilde{X}^T{}_{i,s} + R_{N,i} \right)^{-1} (Y_s - X_{i,s})$$

$$= A_i \frac{1}{S_i} \sum_{s \in B_i} M_{\alpha,i,s} \quad (14)$$

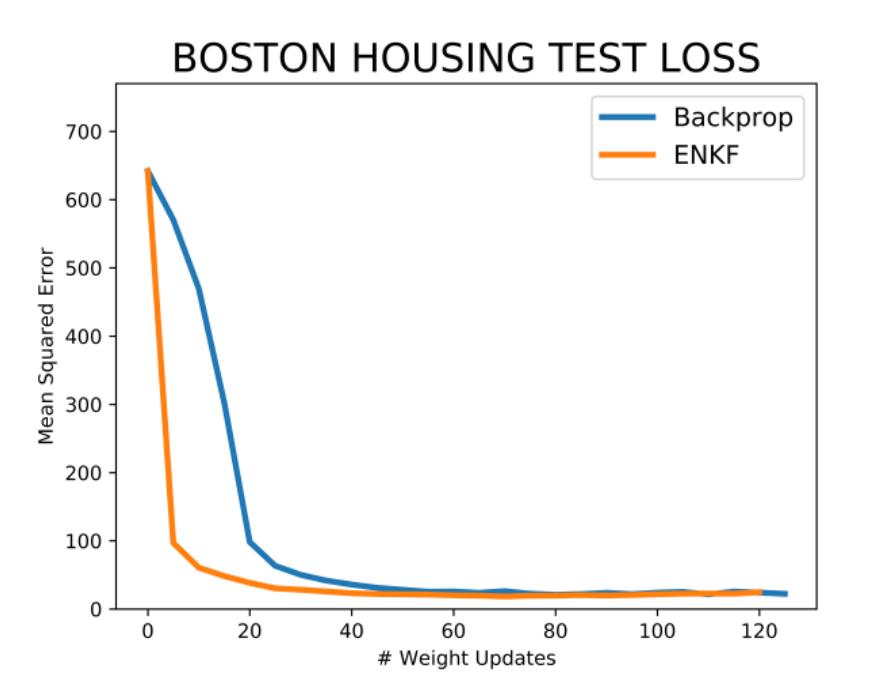
The interpretation of "observational noise" $R_{N,i}$ is the tolerance with which the label or output must be learned.

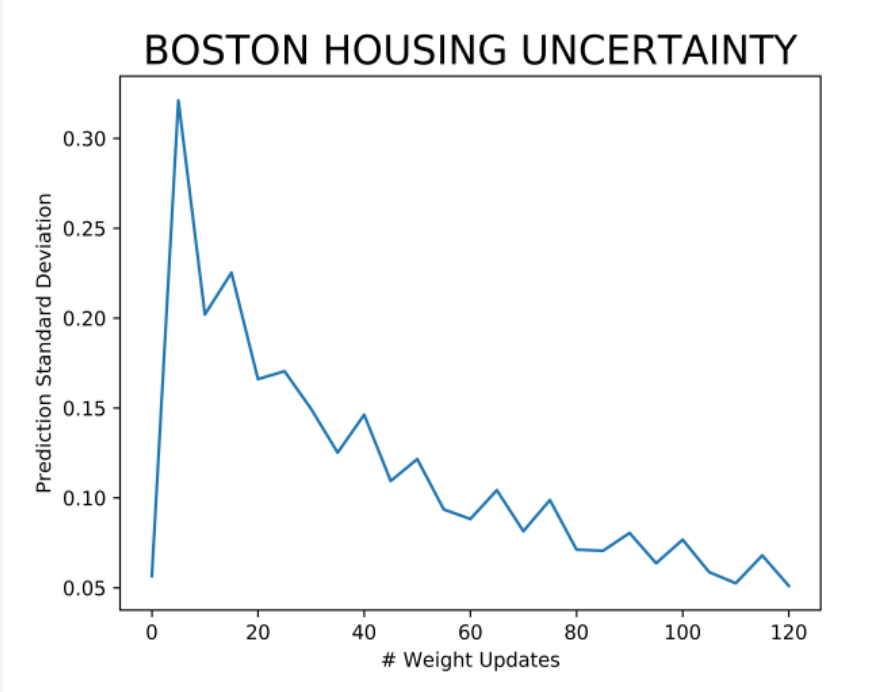
We use an adaptive version where the $R_{N,i}$ is reduced over iterations.

- Simple update rule, many variants possible
- Parallelizable – parameter updates do not require backpropagation
- Quantifies Uncertainty
- Enables Information Gain Assessment

Also see: **arXiv:2008.09915 for application to smoothers and control**

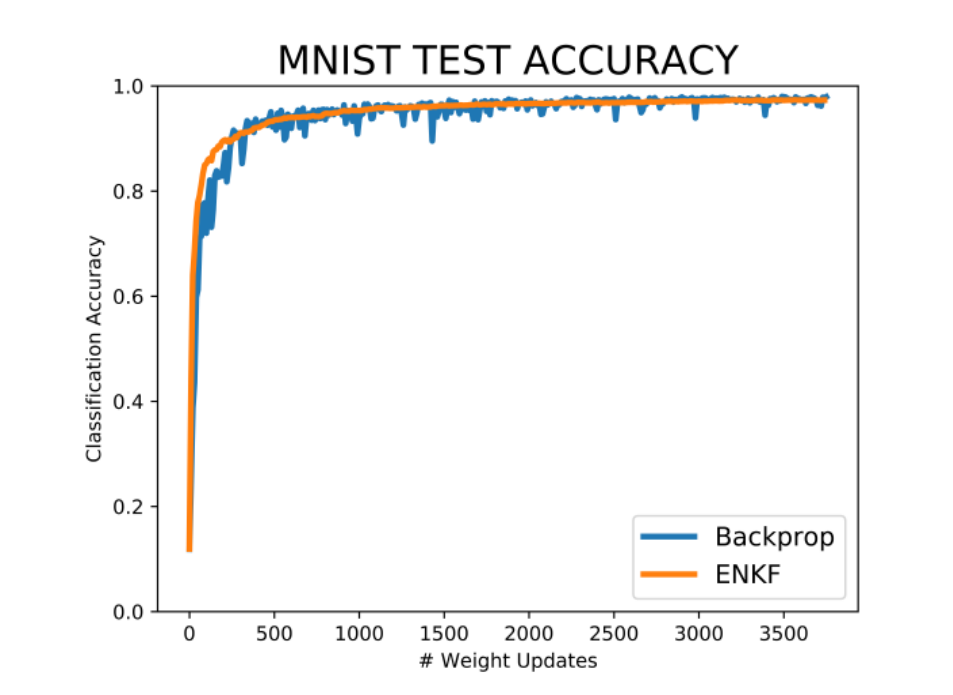
# Boston Housing Example

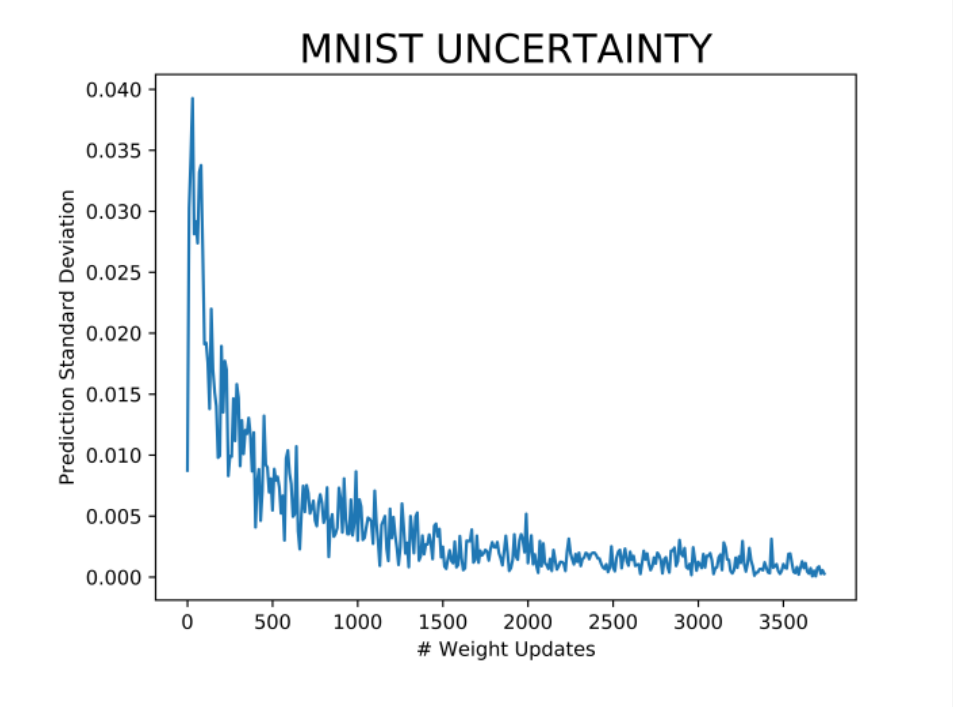



Ensemble Filter

# MNIST Example



# Information Gain for Variable Selection

Quantify Pairwise Mutual Information

$$I(\mathcal{A}:\mathcal{E}) := [\ \Psi_{i,j}\ ]_{m\times n}\,(\mathcal{A}:\mathcal{E}),$$
$$\Psi_{i,j}(\mathcal{A}:\mathcal{E}) := -\frac{1}{2}\ln\left(1-\rho^2(\mathcal{A}_i,\mathcal{E}_j)\right).$$

Sort pairwise mutual information in decreasing order

$$\Psi_l^* \geq \Psi_{l+1}^*,\ 0<l<mn,$$
$$\Psi_k^* := \Psi_{i_k,j_k},\ 1\leq i_k\leq m, 1\leq j_k\leq n.$$

Greedy $\ell_0$ often outperforms $\ell_1$
Proposed approach is exceedingly fast

Forward Variable selection better than

- Monte Carlo or naïve MCMC
- Iterating and eliminating small weights e.g., using $\ell_1$

Informative approach: accelerate optimization $(\mathrm{e.g.}\,\ell_1)$ with information gain

Greedy $\ell_0$ Optimization for Variable Selection

$$k^* = \arg\min_k \underbrace{\sum_{l=1}^{k}\left[1-\frac{1}{\Psi\#}\Psi_l^*\right]}_{\text{Decreasing}} + \underbrace{C(k)}_{\text{Increasing}}$$

# Structure Learning

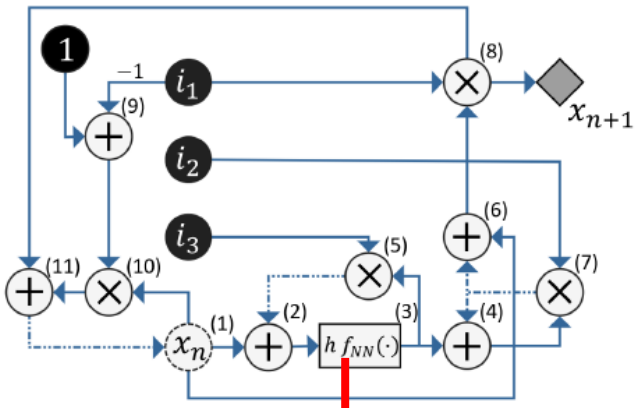
- Learn the optimal structure of the Neural Network from Data
- Many Challenges
  - How do we verify the structure is optimal?
  - Is this Neural Network generalizable
  - Does it extrapolate?
  - How to interpret the neural network
- Possible Solution
  - Neural Networks for Polynomial Dynamics

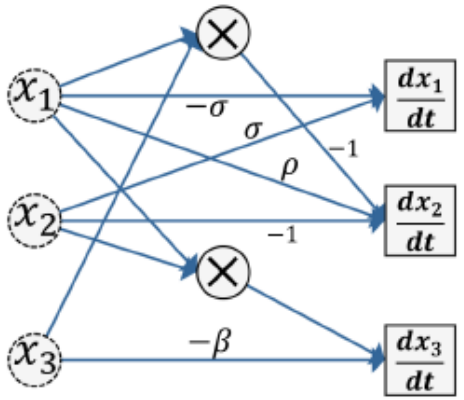
# Polynomial Dynamical Systems Have Exact Neural Circuits

$$\dot{x}_1 = \sigma(x_2 - x_1),$$
$$\dot{x}_2 = \rho x_1 - x_2 - x_1 x_3,$$
$$\dot{x}_3 = -\beta x_3 + x_1 x_2.$$

Lorenz 63 Equations

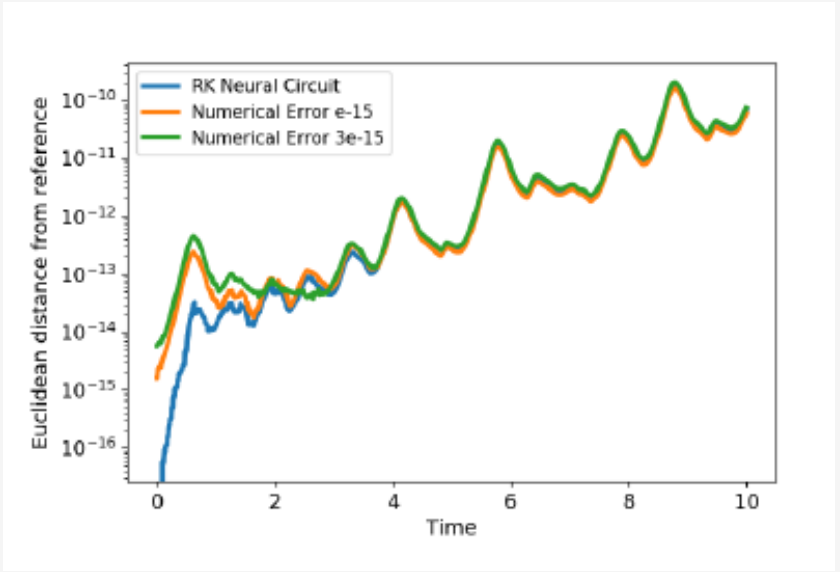
RNN for Runge Kutta




Neural Network for L63 (trivial)

Exact Solution up to numerical accuracy




Use of the PolyNet construct to validate structure learning methodology – here on L63

Also see: **arXiv:2008.09915 and arXiv:1911.10309**

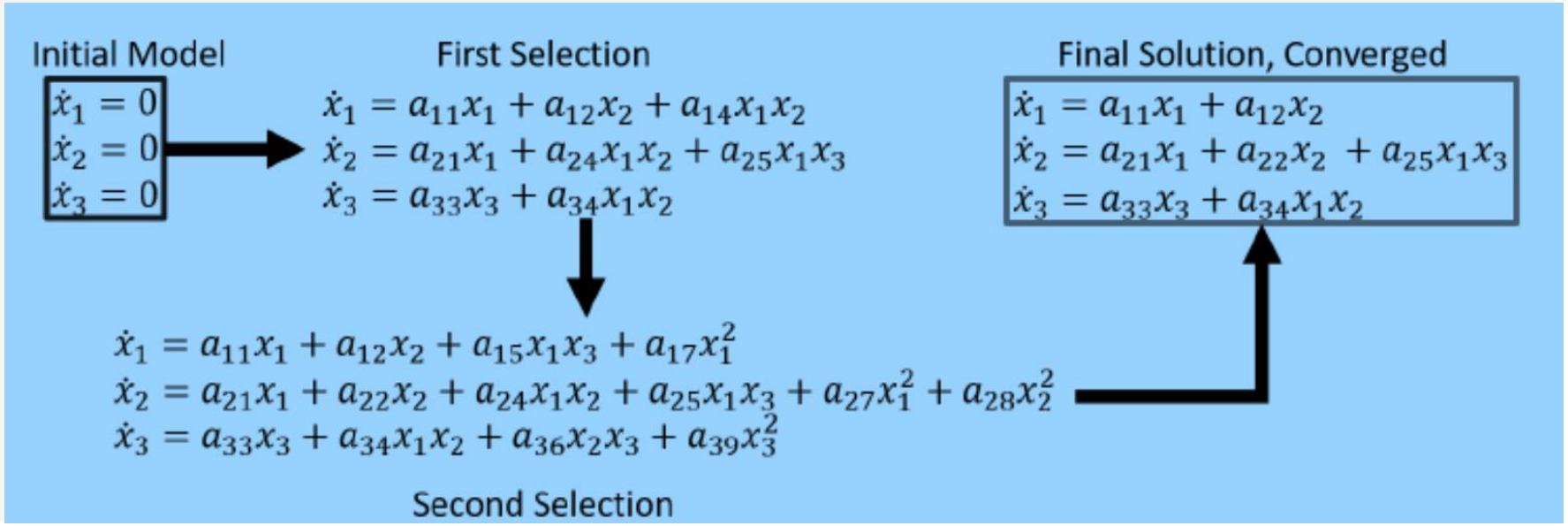
# Informative Ensemble Kalman Learner

Ensemble Kalman Learner Example

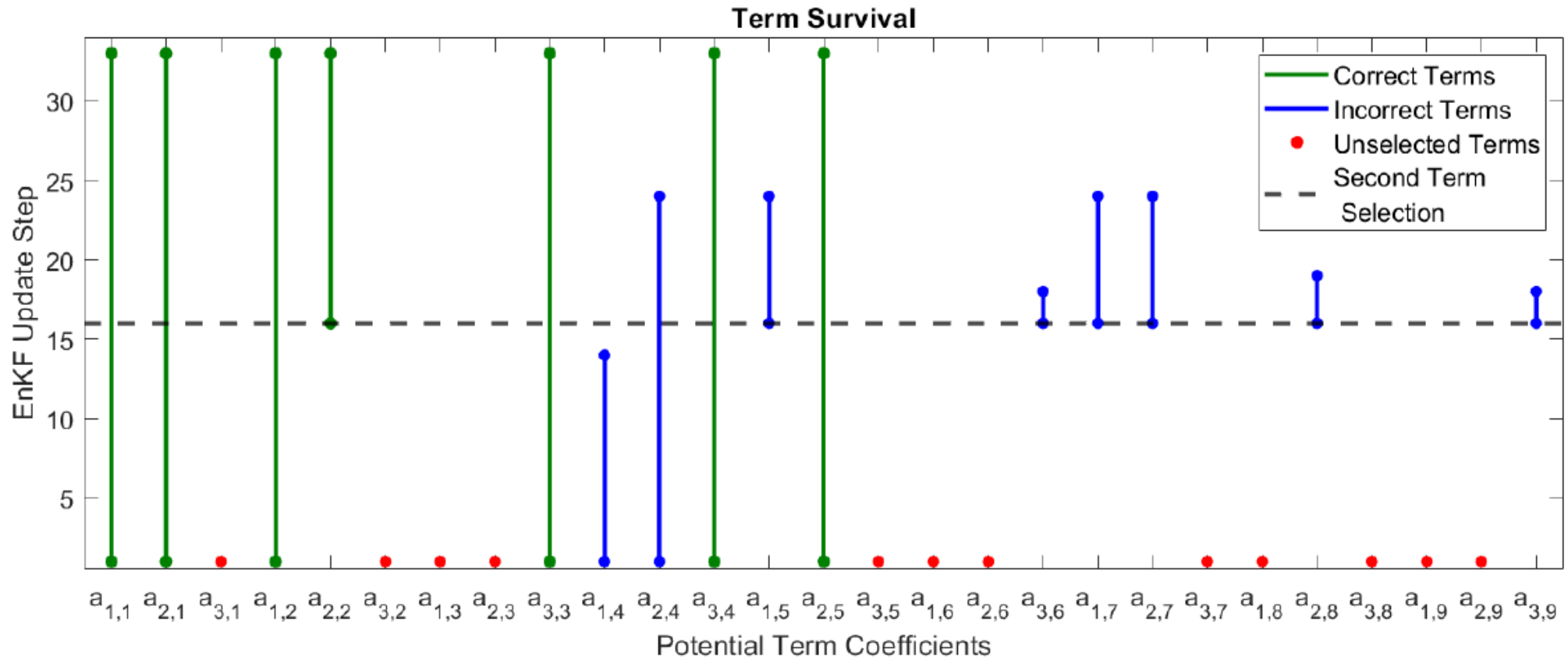
$$\mathbf{X} = (x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3, x_1^2, x_2^2, x_3^2).$$

Ensemble Kalman Learner recovers the equations after 80 iterations, initialized with degree 2 polynomial. Convergence slows as the initial set grows!

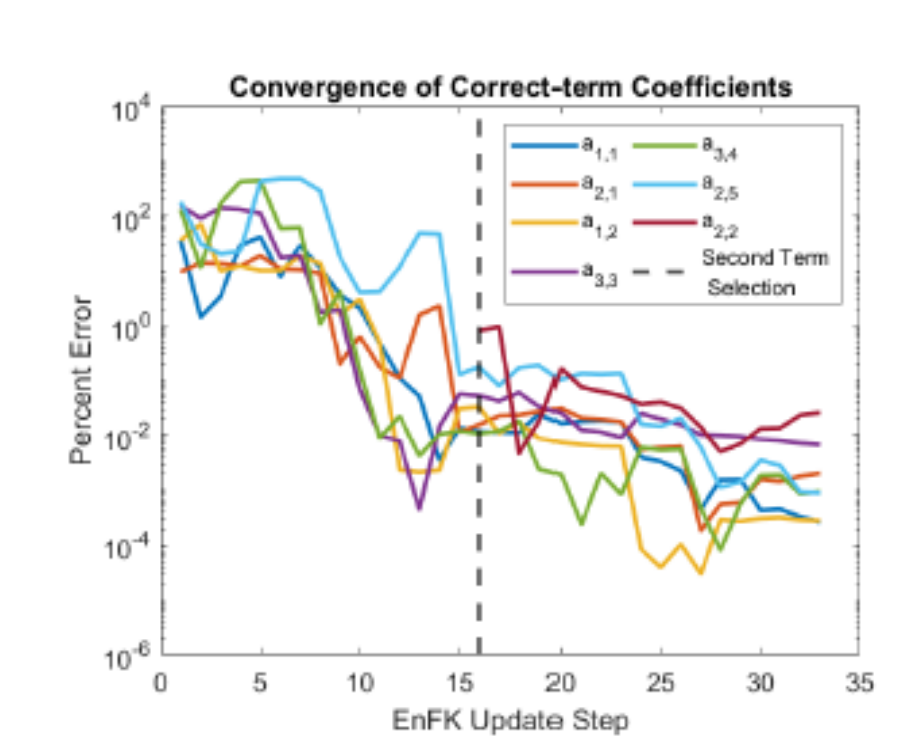
Informative Ensemble Kalman Learner  Recovers equations faster, ***ab initio – 60%***

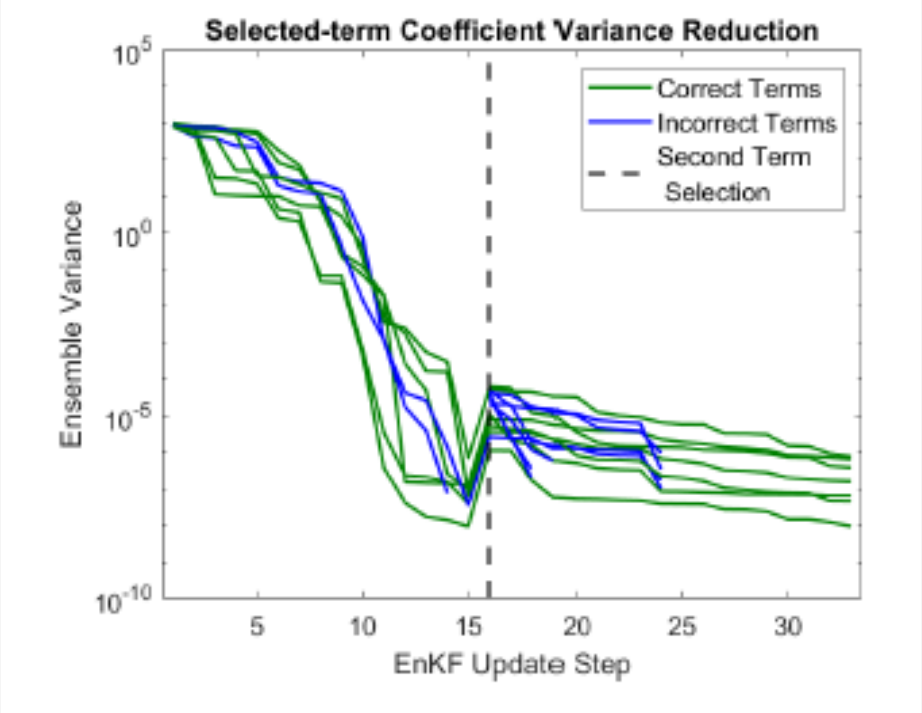


# Survival Chart Lorenz-63



# Convergence



# Summary

- Dynamical Systems are associated with Learning
- Learning is a two-point boundary value problem
- The dynamics of learning is stochastic
- Stochastic Dynamics allows for Information Gain to be quantified
- Ensemble Methods approximate the Fokker Planck
- Informative Ensemble Kalman Learning Efficaciously Recovers Structure
- Poly Nets exactly model discrete time polynomial dynamics and thus correctness is verified.

# The Stochastic Dynamics of Learning

Mini-batch

$$\overline{\nabla J(\alpha_i)} = \frac{1}{|B_i|} \sum_{s \in B_i} \frac{\partial J\left(\alpha_i;\ [x_1, y_N]_s\right)}{\partial \alpha_i},$$

Ito Equation

$$d\alpha_t = \mu_t(\alpha_t)\ dt + \sigma_t(\alpha_t)\ \eta_t.$$

Perturbation Expansion

$$-\tau \overline{\nabla J(\alpha_i)} = \mu(\alpha_i) + w(\alpha_i),$$

Fokker Planck

$$\frac{\partial p_\alpha}{\partial t} = -\sum_{j=1}^{n} \frac{\partial}{\partial \alpha_j} \left[\mu_{t,j}\ p_\alpha\right] + \sum_{k=1}^{n} \sum_{l=1}^{n} \frac{\partial^2}{\partial \alpha_k \partial \alpha_l} \left[D_{t,kl}\ p_\alpha\right]$$

Stochastic Model

$$w(\alpha_i) = \sigma(\alpha_i)\ \eta_i,$$

$$\mathcal{I}(\mathcal{A}_t : \mathcal{D}|\mathcal{A}_0) \quad \mathcal{I}(\mathcal{D}_{s,t'>t} : \mathcal{E}_t|\mathcal{D})$$

$$\mathcal{I}(\mathcal{S}_{s,t'} : \mathcal{E}_t|\mathcal{S}_t) \quad \mathcal{I}(\mathcal{A}_{s,t'} : \mathcal{E}_t|\mathcal{A}_t)$$